



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/539,090	03/30/2000	Gueorgui B. Chkodrov	202409	8897

7590 05/17/2004
Leydig Voit & Mayer LTD
Two Prudential Plaza
Suite 4900
180 North Stetson
Chicago, IL 60601-6780

EXAMINER

KENDALL, CHUCK O.

ART UNIT PAPER NUMBER

2122

DATE MAILED: 05/17/2004

5

Please find below and/or attached an Office communication concerning this application or proceeding.

8

Office Action Summary

Application No.

09/539,090

Applicant(s)

CHKODROV ET AL.

Examiner

Chuck Kendall

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 March 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the application filed 03/30/00.
2. Claims 1-24 have been examined.

Claim Rejections - 35 USC § 103

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1 – 10, 12 – 24 are rejected under 35 U.S.C. 102(b) as being anticipated by Rose et al. USPN 5,790,861 (hereinafter Rose).

Regarding claims 1, 8 & 16, Rose anticipates a method (15:37), a computer readable medium (18:23), and of constructing a computer program developed with an object-oriented programming language, the method comprising: declaring a base class as replaceable in a first source file (see Header file, FIG.2, 295,200, also refer to, 6:61-67) invoking, in the first source file, an operator to create an object of the base class (FIG.2, 200, Header file); compiling the first source file into a first module, including emitting an instruction to create an object of the base class in response to the invoking of the operator to create in the first source file (FIG.2, 210,220 compiled Header file); defining a replacement class inheriting from the base class in a second source file (FIG.2,242, also refer to, 6:31-34); instructing in the second source file to replace the base class with the replacement class to cause creation of an object of the replacement class when the instruction in the first module to create an object of the base class is executed (FIG.2, 244) during program execution (FIG.2, steps 280 , 290 and 295 and associated text);

compiling the second source file into a second module (FIG.2, 250); and combining the first and second modules in an executable program (FIG.2,270).

As per claims 2, 9,10 and 17, method' as in claim 1, wherein the object-oriented programming language is the C++ language (6:3).

As per claims 3,14,15 a method as in claim 2, wherein the step of declaring the base class as replaceable includes defining a virtual constructor of the base class (9:46-50).

As per claim 4, a method as in claim 2 , further including declaring the replacement class as a replacement of the base class by defining a constructor of the replacement class for replacing the base class (9:62-65, see c++ construct and accessorize).

As per claim 5, a method as in claim 1, wherein the step of compiling the second source file includes emitting in the second module an instruction to generate a class replacement record indicating that the replacement class is a replacement of the base class (FIG. 2, 295).

As per claim 6, a method as in claim 5, wherein the step of compiling the first source file includes emitting an instruction in the first module to search for a class replacement record concerning the base class (12:10-13, see query and repository).

As per claim 7, a method as in claim 6, wherein the step of compiling the first source file includes emitting creation information for an object of the base class in the first module, and wherein the step of compiling the second source file includes emitting creation information for an object of the replacement class in the second module (FIG.2, 200, 240).

As per claim 12, a computer-readable medium as in claim 8, wherein the second portion of the source code further includes an instruction to end replacement of the base class with the replacement class (9:60-65, see accessorize and not accessorize).

As per claim 13, a computer-readable medium as in claim 8, wherein the second portion of the source code further includes an instruction to use a function containing the instruction in the first portion of the source code to create an object of the base class (6:25-35).

Regarding claim 18, Rose anticipate:; a computer-readable medium having computer executable instructions and data comprising:

creation information for creating an object of a base class (FIG.2, 295);

creation information for creating an object of a replacement class derived from the base class (FIG.2,200 240, 295);

an instruction to register a replacement relationship between the base class and the replacement class (9:65-67);

an instruction to determine whether a registered replacement relationship between the base class and the replacement class exists upon receiving a request to create an object of the base class (FIG.2, 290, see YES or NO);

an instruction to access the creation information for the replacement class upon a determination of existence of the registered replacement relationship between the base class and the replacement class (9:60-67, &10:3-7);

an instruction to create an object of the replacement class using the creation information for the replacement class in response to the request to create an object of the base class (FIG.2, 290, 295).

As per claim 19, a computer-readable medium as in claim 18, wherein the instruction to create includes an instruction to call a virtual constructor of the replacement class identified in the creation information for the replacement class (9:46-50).

As per claim 20, a computer-readable medium as in claim 18, further including an instruction to unregister the replacement relationship between the base class and the replacement class (9:60-65, see accessorize and not accessorize).

As per claim 21, a computer-readable medium as in claim 18, further including an instruction to call a virtual destructor of the base class to delete the object of the replacement class (9:46-50).

Regarding claim 22, Rose anticipates a computer-readable medium having computer executable instructions to performs steps for compiling a source listing in an object-oriented programming language into an executable module, comprising:

upon reading a statement in the source listing defining a first class as replaceable, emitting into the executable module a creation information block for the first class (FIG.2, 295);

upon reading a statement in the source listing defining a second class inheriting from a first class as a replacement for the first class, emitting into the executable module a creation information block for the second class, upon reading an instruction in the source listing to replace the first class with the second class, emitting an instruction to store a class replacement record for the first and second classes in a class replacement registration list (10:60 – 63) and;

upon reading an instruction in the source listing to create an object of the first class, emitting into the executable module:

an instruction to search the class replacement registration list (12:10-13, see query and repository) and to return the pointer to the creation information block for the second class if a class replacement record for the first and second classes is found and otherwise return the pointer to the creation information for the first class (8:16, see base class, pointer also see 8:47-51);

an instruction to create during program execution, (FIG.2, steps 280, 290 and 295, shows running the program and modifying if need be) an object according to the creation information block pointed to by the returned pointer (8:47-51).

As per claim 23, a computer-readable medium as in claim 22, including further computer-executable instructions to perform the step of emitting a virtual destructor for the first class for deleting the object created (9:46-50).

As per claim 24, a computer-readable medium as in claim 22, wherein the object-oriented programming language is the C++ language (6:3).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rose et al. USPN 5,790,861 (hereinafter Rose), as applied in claim 8 in view of, Lewallen USPN 6,385,769 B1.

Regarding claim 11, Rose discloses all the claimed limitations as applied in claim 8 above. Rose doesn't explicitly disclose wherein the object-oriented programming language is an interpreted language. However, Lewallen does disclose this feature in a similar configuration using Java, which is an interpretive language (7:52-55). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Rose and Lewallen because, Java is the preferred OOP language for internet and cross platform use, and is operating system independent.

Response to Arguments

7. Applicant's arguments filed 03/08/2004 have been fully considered but they are not persuasive to overcome previous rejection. Below are Examiners response to Applicants argument.

Argument (1), Applicant on page 7, asserts that Rose doesn't teach "replacement during program execution".

Response (1), Examiner believes that Rose does teach this limitations. In Rose FIG.2, steps 280, 290 and 295, Rose shows being able to modify and edit the previous base class file (Header) to produce a modified version which Examiner interprets as replacing during program execution.

Also Applicant argues for an unclaimed merit of distinction in independent claims. Applicant argues that Rose doesn't teach allowing replacement of code to be undone during program execution. There is no where in the independent claims 1, 8 or 22 that

claims this, therefore Applicants argument is moot. Examiner would also like to point out to Applicant that the plain language of the claims do not exclude any reading from "recompiling" and/or "relinking" steps.

Conclusion

8. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Correspondence Information

8. Any inquires concerning this communication or earlier communications from the examiner should be directed to Chuck O. Kendall who may be reached via telephone at (703) 308-6608. The examiner can normally be reached Monday through Friday between 8:00 A.M. and 5:00 P.M. est.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam *can be* reached at (703) 305-4552.


Art Unit: 2122

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

For facsimile (fax) send to 703-7467239 official and 703-7467240 draft

Chuck D. Kendall

Software Engineer Patent Examiner
United States Department of Commerce

A handwritten signature in black ink, appearing to read 'Tuan Dam', with a long horizontal stroke extending to the right.

**TUAN DAM
SUPERVISORY PATENT EXAMINER**